We present and analyze the semantics and static type system for BeCecil, a theoretical (core) language with multimethods. BeCecil is a simple and orthogonal version of object-oriented languages like Cecil, CLOS, and Dylan. BeCecil has a new, simple mechanism for information hiding, which allows subclassing and yet can preserve representation invariants. BeCecil is also block-structured; within a block, one can extend a generic function with new multimethods, which may come from other generic functions. The inheritance relationships of objects may be extended in any block, and are statically scoped. The type system separates classes from types, and inheritance from subtyping. Subtype relationships are also extensible and statically scoped. These features combine to make BeCecil unusually expressive, while still allowing static typechecking.

However, most exploratory programming languages (and language environments) only allow a single programmer to interact with the code and data being developed; if a team of programmers wishes to collaborate, they must use project management and revision control tools which greatly discourage or hinder an exploratory style. Craig Chambers and Gary Leavens. BeCecil, a core object-oriented language with block structure and multimethods: Semantics and typing. In proceedings of the The Fourth International Workshop on Foundations of Object-Oriented Languages (FOOL 4), Paris, France, January 1997. DHW98. John C. Doppke, Dennis Heimbigner, and Alexander L. Wolf. Structured Programming. Structured programming evolved in the 1960s and 1970s. The name refers to the practice of building programs using a set of well-defined structures: Sequence structure defines the default control flow in a program. This structure is built into programming languages. A computer executes lines of code in the order in which they are written. Repetition structures (or looping structures) use the loops. In a repetition structure, the program checks a conditional statement and executes a loop based on the condition. If the condition is true, then a block of one or more commands is repeated until the condition is false. Object-Oriented Programming. In the 1980s object-oriented programming (OOP) was developed. The concept of an object-oriented database programming language (OODBPL) is appealing because it has the potential of combining the advantages of object orientation and database programming to yield a powerful and universal programming language design. A uniform and consistent combination of object orientation and database programming, however, is not straightforward.